Curso de Programação PYTHON

No site Curso em Video tem a explicação para o que serve o Python, instalção do Python e Idle no computador, instalação da ferramenta Pycharm que serve para fazer os programas com mais facilidade e programa Qpython3 no celular sistema android.
O que está nessa apostila é o curso em Python e não ensina instalar nenhuma ferramenta.

Aula 1 – introdução ao mundo da programação – video no site
Aula 2 – Para que serve o python – video site
Aula 3 – instalando Python3 e Idle
e **Aula 5** – instalando Pycharm e Qpython 3 - Nesta aula, veremos como instalar e configurar a IDE (Integrated Development Environment) Python chamada PyCharm no Windows, MacOS e Linux. Além disso, veremos como programar Python no Android, utilizando o QPython3. - video no site.

Video aula 4: primeiros comandos em python3 – video no site
Base Teórica:

No Python para é necessário utilizar aspas simples para a grande maioria de mensagens, por exemplo: 'Olá Mundo'

No Python todos os comandos são funções, sendo que toda a função tem que ter parênteses, exemplo: ('Ola Mundo !') - esse parênteses podem ser usadas com qualquer função.
Print é uma função para ser utilizada que significa escreva. Por exemplo: Você esta pedindo para escrever algo na tela :

print('Olá Mundo !') = Olá Mundo !

Isso é uma programação básica que é necessário aprender em todas as linguagens de programação.
Números não utilizam aspas, por que eles não são mensagens, a diferença entre mensagens e números é que primordialmente mensagens são utilizadas para aparecer mensagens na tela e números são utilizado primordialmente para fazer cálculos.
Se o programador quiser fazer uma conta tem que ser sem aspas, entre dois números pode ser colocado uma operação, por exemplo: 7 + 4
Se o programador quiser escrever esse número na tela, é necessário fazer a função print, por exemplo:

print(7 + 4) = que irá aparecer 11

print(‘Olá Mundo !) - Função de texto utilizando aspas, porque é uma mensagem.

print(7 + 4) – Função de texto que não é uma mensagem e sim números não utilizam aspas. Se oe programador utilizar aspas nos números, por exemplo:
print(‘7’ + ‘4’) – não vai dar erro, mas essa função não irá somar e sim juntar os números, o exemplo vai mostrar 74 e não a soma dos dois números.
Nesse caso no Python pode ser o símbolo de + ou pode ser uma , existe diferenças, tem casos que a virgula vai funcionar melhor e tem casos que o + vai funcionar melhor.
Se o programador for utilizar uma mensagem com o número é necessário utilizar a virgula.

Utilizando Variáveis , exemplo:

Atenção: Variáveis, escrever sempre com letras minúsculas.

Variáveis - nome
idade - Toda variável é um objeto, para o Python.
peso

Toda variável pode receber valores colocando o sinal de = que significa recebe.
Exemplo:

nome =
idade =
peso =

Cada variável tem um valor, por exemplo:

nome = ‘Guanabara’
idade = 25
peso = 75.8

E para mostrar esses três valores na tela pode ser utilizado a função print, por exemplo:

print(nome, idade, peso) – nesse caso não pode utilizar o + e sim a , (virgula), porque o nome é uma mensagem e idade e peso são números, no caso idade é um número sem virgula e peso tem virgula.
Variáveis são espaços na memória aonde você pode guardar itens.

Para fazer uma interatividade com o usuário é necessário colocar a função input(significa ler), veja o exemplo:

```
nome = input(‘Qual é o seu nome? ‘)
idade = input(‘Quantos anos você tem?)
peso = input(‘Quanto você pesa?)
print(nome,idade,peso)
```

Digitando os comando acima, no terminal vai aparecer como o exemplo abaixo:

```
Qual é o seu nome? Guanabara
Quantos anos você tem? 25
Quanto você pesa? 75.8
Guanabara 25 75.8
```

Para não precisar ficar digitando todos os comando é possível criar scripts específicos.

O modo interativo é quando o programador quer testar algo, que seria no próprio terminal ou no idle do python, se o programador quiser criar um programa vai para o modo script no idle, abrindo file, new file e pode salvar o script no save, depois que salvar o script para ver o que a programação faz é só clicar no Run, Run module.

Vamos fazer 3 desafios pelo Idle do Python, utilizando o que já foi explicado, respostas vai estar no final da apostila:

Desafio 1

Crie um script python que leia o nome de uma pessoa e mostre uma mensagem de boas vindas de acordo com o valor digitado, é necessário aparecer exatamente como o exemplo abaixo:

```
Qual é o seu nome? Gustavo
Olá Gustavo! Prazer em te conhecer .
```

Desafio 2

Crie um script Python que leia o dia, mês e ano de nascimento de uma pessoa e mostre uma mensagem com a data formatada:

```
Dia = 17
Mês = Março
```

Ano = 1978

Você nasceu no dia 17 de março de 1978. Correto?

Desafio 3

Crie um script Python que leia dois números e tente mostrar a soma entre eles:

Primeiro número: 6
Segundo número: 3
A soma dos dois 9

Este desafio 3, para que ele execute de forma correta é necessário colocar INT, porque colocando somente o input, o programa irá juntar os números e não somar os dois. Utilizando o comando INT seguindo do comando input, sendo que o comando input é necessário ficar entrer parênteses junto ao restante das escritas.
Este comando vai ser ensinado em breve, tente fazer o programa, respostas na ultima página.

No 1° exercício ensina como começar um novo projeto no Pycharm.

Faça os exercícios conforme explicado os primeiros passos acima.

No Python 3 existe a saída formatada, que é usada na função print, colocando o símbolo {} na fraze e escrevendo .format(), dentro dos parênteses colocar as variáveis que irão mostrar as respostas no print, exemplo abaixo:

```
nome = input('Qual é seu nome? ')
Qual é seu nome? David
print('Prazer em te conhecer {}!'.format(nome))
Prazer em te conhecer David!
```

Os exercícios irão ter exemplo do que se deve aparecer na tela no momento que o programa for executado, faça aparecer da mesma forma que o exemplo do exercício estiver mostrando, o mais importante é fazer ele executar exatamente o que está pedindo, se você seguir o passo a passo das explicações do curso. Lembrando também que, não quer dizer que o programa esteja funcionando que ele esteja correto, então preste bem atenção e veja se os resultados estão corretos.

Exercício 1

Escreva um programa que apareça 'Olá Mundo!' na tela:

Olá Mundo !

```
print('Olá Mundo !')
```

Exercício 2

Faça um programa que leia o nome de uma pessoa e mostre uma mensagem de boas vindas:

o nome pode ser de qualquer pessoa, abaixo é apenas um exemplo do que deve aparecer na tela:

Qual é seu nome? David
Prazer em conhecer você David

```
nome = input('Qual é seu nome? ')
print('Prazer em conhecer você', nome)
```

Aula 6 – Tipos Primitivos e Saída de Dados:

Nessa aula, vamos aprender como funcionam os tipos primitivos no Python e as peculiaridades do int() float() bool() e str(). Além disso, veremos como fazer as primeiras operações com a função print() do Python.

Tipos Primitivos:

Toda linguagem de programação trabalha com 4 tipos primitivos, o Python existe mais tipos primitivos.

INT – Número inteiro, número positivo, nagativo ou nulo são valores inteiros para o python. Ex:(7, -4, 0,0.9875)

Float – Número real, número que existem pontos que na linguagem de programação significa flutuante. Ex:(4.5, 0.076, -15.223, 7,0)

Bool – Aceita somente dois valores, sempre que for utilizar o valor booleano, que é um valor lógico é necessário colocar dessa forma: True ou False, ambos com a 1° letra maiúscula e o restante das letras tem que ser minúscula, nesse caso você estaria colocando a primeira letra capitalizado de forma maiúscula. Este tipo primitivo vai mostrar se o que você escreveu é verdadeiro ou falso.

Str – string utiliza palavras e números que estão entre aspas(ex: ‘ola’, ‘7,5’).
Abre aspas e fecha aspas sem nenhum conteúdo é uma string vazia: exemplo: print(‘’). Todas as palavras tem que estar sobre aspas.

Fazendo print com mais recursos:

{} - Chaves saída formatada.

Por exemplo:

Jeito simples: print(‘A sala vale’, s)

sintaxe formatada ex: print(‘Asoma vale {}’.format(s))

para saber o tipo primitivo de uma programação:
ex:

```
n1 = input(‘Digite um valor:’)
print(type(n1))
```

```
Digite um valor : 6
<class ‘str’>
```

Na programação o tipo primitivo de uma variável tem que ser expecificado no momento em que for utilizada.
Neste exemplo acima, você colocando somente o input, o programa vai falar que o tipo primitivo é sempre string ‘str’, se colocar outra classe na frente por exemplo int, ele irá falar que a classe é int, ex:

```
n1 = int(input('Digite um valor: '))
print(type(n1))
```

```
Digite um valor: 6
<class 'int'>
```

Juntar uma string na outra se chama concatenação, ex:

```
n1 = input(‘Digite um número: ‘)
n2 = input(‘Digite outro número: ‘)
print(‘Juntando o numero {} e {}, formam o número {}’.format(n1, n2, n1 + n2))
```

Digite um Número: 6
Digite outro Número: 2
Juntando o número 6 e 2, formam o número 62

Dessa forma eles não estão somando e sim concatenando(juntando).

Exemplo de exerçício com o print formatado:

Digite o valor : 6
Digite o valor: 2
A soma entre 6 e 2 vale 8

o formato do print tem que ser escrito da forma atual ex:

```
n1 = int(input('Digite o valor: '))
n2 = int('Digite outro valor: '))
s = (n1 + n2)
print('A soma entre {} e {} vale {}.'.format(n1,n2, ,s))
```

Fazendo desta forma, vai mostrar como o exemplo acima.

Para ver métodos de tipos é necessário usar o is, por exemplo:

```
n1 = input('Primeiro número: ')
print('É um número? ', n1.isnumeric())
```

vai escrever da forma abaixo:

Primeiro número: 3
É um número ? True

Os IS vai mostrar se é possível converter o valor da variável com o tipo primitivo que está antes do input, ou mesmo se estiver somente input e você digitar um número e você perguntar se é número, o programa vai falar que é verdadeiro.
Existem vários métodos de tipos is para fazer testes de tipo primitivo, sendo que se a variável tiver somente input, ele sempre será do tipo string 'str':

isnumeric – é um número?
isalpha – é alfabético?
isalnum – é alfabético e número?
isdecimal-
isdigital-

isindentifier-
islower – está somente em letra minúscula.
isprintable -
isspace – somente espaços
istitle – se a 1? letra ‘maúscula é um titulo.
isupper – está somente com letras maiúsculas

Exercício 3 – Somando dois números

Crie um programa que leia dois números e mostre a soma entre eles:

Digite um número: 5
Digite mais um número: 5
A soma de 5 e 5 é 10.

```
n1 = int(input('Digite um número: '))
n2 = int(input('Digite mais um número: '))
print('A soma de {} e {} é {}.'.format(n1, n2, n1 + n2))
```

Exercício 4 – Dissecando uma Variável

Faça um programa que leia algo pelo teclado e mostre na tela o seu tipo primitivo e todas as informações possíveis sobre ele:

Digite algo: Programador
O tipo Primitivo é <class 'str'>.
É um número? False
Está em Maiúscula? False.
Está em Minúscula? False.
É um Título? True.
É alfanumérico? True.
Tem somente espaços? False.
É alfabético? True.

```
frase = input('Digite algo: ')
print('O tipo Primitivo é {}.'.format(type(frase)))#Para saber o tipo Primitivo tem que colocar type()
print('É um número? {}'.format(frase.isnumeric()))
```

```
print('Está em Maiúscula? {}.'.format(frase.isupper()))
print('Está em Minúscula? {}.'.format(frase.islower()))
print('É um Título? {}.'.format(frase.istitle()))
print('É alfanumérico? {}.'.format(frase.isalnum()))
print('Tem somente espaços? {}.'.format(frase.isspace()))
print('É alfabético? {}.'.format(frase.isalpha()))
```

Aula 7 – Operadores Aritméticos – Video no site

Nessa aula, vamos aprender quais são os operadores aritméticos do Python e também sua ordem de precedência dentro de expressões matemáticas. Veja como funcionam os operadores de adição, subtração, multiplicação, divisão, exponenciação e quociente na linguagem Python.

Teoria:

```
n1 = int(input('Digite um número: '))
n2 = int(input('Figite outro número: '))
s = n1+ n2
print('A soma vale {}.'.format(s))
```

neste exercício foi utilizado o operador de adição.

Operador aritmético simples:

+ Adição
- Subtração
* Multiplicação
/ Divisão

Operadores de exponenciação ou potência:

** Potência
// Divisão inteira
% módulo ou resto da divisão

Para o python e para a maioria da linguagem de programação o simbolo de % não calcula porcentagem, o simbolo de % é utilizado para o módulo que é o resto da divisão.
Quando aparecer dois simbolos de igual == é quando o operador quer testar se o item é igual ao outro, quando fica somente = significa que recebe e == significa igual:

ex:

5 + 2 == 7
5 – 2 == 3
5 * 2 == 10
5 / 2 == 2.5
5 ** == 25
5 // 2 == 2
5 % 2 == 1

Ordem de precedência:

Quais são os operadores que são executados em primeiro lugar:

1° ( )
2° **
3° * / // %
4° + -

exemplo:

5 + 3 * 2 == 11 – primeiro é somado a multiplicação * para depois somar a adição +
3 * 5 + 4 * * 2 == 31 - o primeiro somado é potência **, depois a multiplicação * e por ultimo a adição +

Se no caso a adição estiver entre ( ) ele será somado primeiro seguido da potência ** e depois multiplicação * como o exemplo abaixo:

3 * (5 + 4) ** 2== 243 primeiro a adição (+) depois potência, seguido da multiplicação *
Para fazer conta de raiz quadrada é necessário colocar(1/2).
Ex: 81 ** (½) = 9.0

para fazer conta de raiz cubica é necessário colocar (1/3)
ex: 127 ** (1/3) = 5.02

Para fazer conta ao cubo pode ser colocado ** 3 ou pow(n, 3) e calculando raiz quadrada pow(n,(1/3))
ex: 4 ** 3

** 2 – ao quadrado
** 3 - ao cubo

para calcular com a função pow() de potência em vez de colocar a potência ** tem que colocar como o exemplo abaixo:

modo normal: n ** (½)
modo pow : pow(n, (1/2))

No print para deixar o valor da soma com duas ou mais casa decimais é só colocar {:.2f} a soma mostrará somente duas casas decimais.
Para não quebrar a de um print para o outro é só colocar end=’’ no 1° print.
SE quiser quebrar a linha no meio do print é só colocar a letra \n.
O print a partir da atualização 3,0 é uma função interna do python.
Para fazer alinhamento a esquerda ou a direita é só colocar dentro das chaves o sinal < >, por exemplo: {:<20}.
Para deixar a palavra centralizada é só utilizar o sinal ^, por exemplo:{:^20}
para colocar símbolos na frente é só colocar na frente do ^, por exemplo: {:=^20}

importante! As variáveis no python podem ter acento exemplo: (média), mas é melhor não colocar.

Exercício da aula 7:

Exercício 5 – Antecessor e Sucessor

Faça um programa que leia um número Inteiro e mostre na tela o seu sucessor e seu antecessor:

Digite um número: 2
O número que você digitou é 2, o seu antecessor é 1, e seu sucessor é 3

```
n = int(input('Digite um número: '))
print('O número que você digitou é {}, o seu antecessor é {}, '
      'e seu sucessor é {}'.format(n, n - 1, n + 1))
```

Neste exercício você pode fazer com variáveis para o antecessor e sucessor, mas neste caso, o programa é simples, pode ser feito direto como o exemplo acima.
Se você não colocar a variável o programa fica mais leve, mas tem programa que você for fazer pode necessitar das variáveis, que pode ser pedido mais a frente no programa que estiver executando, depende muito do que vai ser executado.

Exercício 6 – Dobro, Triplo, Raiz Quadrada

Crie um algoritmo que leia um número e mostre o seu dobro, triplo e raiz quadrada:

```
Digite um número: 81
O número que você digitou é 81.
O dobro dele é 162
O triplo dele é 243
A Raiz quadrada dele é 9.00
```

```
n = int(input('Digite um número: '))
print('O número que você digitou é {}.'.format(n))
print('O dobro dele é {}'.format(n * 2))
print('O triplo dele é {}'.format(n * 3))
print('A Raiz quadrada dele é {:.2f}'.format(n ** (½)))
```

Este exercício pode ser feito com a função pow(), para calcular a raiz quadrada.
Se quiser fazer somente um print e deixar da mesma forma, uma linha embaixo da outra é só colocar \n, por exemplo:

```
print('O número que você digitou é {} \n O dobro dele é {} \n O triplo dele é {} \
n A Raiz quadrada dele é {:.2f}'.format(n, n * 2, n * 3, n ** (½))
```

Exercício 7 – Média Aritmética

Desenvolva um programa que leia as duas notas de um aluno, calcule e mostre a sua média:

```
Digite a 1° nota do aluno: 7
Digite a 2° nota do aluno: 5.5
A primeira nota do aluno é 7.0
A segunda nota do aluno 5.5
A média de nota deste aluno é 6.2
```

```
nota1 = float(input('Digite a 1° nota do aluno: '))
nota2 = float(input('Digite a 2° nota do aluno: '))
resultado = (nota1 + nota2) / 2
print('A primeira nota do aluno é {}'
    ' \nA segunda nota do aluno {}'
    ' \nA média de nota deste aluno é {:.1f}'.format(nota1, nota2, resultado))
```

Importante! As variáveis no python podem ter acento exemplo: (média)

Exercício 8 – Conversor de Medidas

Escreva um programa que leia um valor em metros e o exiba convertido em centímetros e milímetros:

```
Digite uma distância em mêtros: 3
O valor que você digitou é 3.0M
e convertido em:
300 cm - centimetros
3000 mm - Milimetros
0.03 hm - hectometro
0.003 km - kilometro
0.3 dam - decametros
30 dm -diametros
```

```
m = float(input('Digite uma distância em mêtros: '))
print('O valor que você digitou é {}M \ne convertido em:'.format(m))
cm = m * 100
mm = m * 1000
hm = m / 100
km = m / 1000
dam = m / 10
dm = m * 10
print('{:.0f} cm - centimetros'.format(cm))
print('{:.0f} mm - Milimetros'.format(mm))
Neste exercício o professor converte tambèm:
print('{} hm - hectomêtro'.format(hm))
print('{} km - kilomêtro'.format(km))
print('{} dam - decametros'.format(dam))
print('{:.0f} dm -diametros'.format(dm))
```

Exercício 9 – Tabuada

Faça um programa que leia um número Inteiro qualquer e mostre na tela a sua tabuada:

```
********** Tabuada **********
```

Digite um número: 5

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

```
print('*' * 10, 'Tabuada', '*' * 10)
print('')
n = int(input('Digite um número: '))
print('')
print('{} x {:2} = {}'.format(n, 1, n * 1))
print('{} x {:2} = {}'.format(n, 2, n * 2))
print('{} x {:2} = {}'.format(n, 3, n * 3))
print('{} x {:2} = {}'.format(n, 4, n * 4))
print('{} x {:2} = {}'.format(n, 5, n * 5))
print('{} x {:2} = {}'.format(n, 6, n * 6))
print('{} x {:2} = {}'.format(n, 7, n * 7))
print('{} x {:2} = {}'.format(n, 8, n * 8))
print('{} x {:2} = {}'.format(n, 9, n * 9))
print('{} x {:2} = {}'.format(n, 10, n * 10))
```

se quiser colocar símbolos nos seus exercícios é só colocar como o print Tabuada acima.

Exercício 10 – Conversor de Moedas

Crie um programa que leia quanto dinheiro uma pessoa tem na carteira e mostre quantos dólares ela pode comprar:
considere que U$1.00 custa R$3.27

Quanto dinheiro você tem na carteira? R$19.88
Com R$19.88 você pode comprar U$6.08

```
carteira = float(input('Quantos de dinheiro você tem na Carteira? R$ '))
dolar = carteira / 3.27
```

```
print('O valor que você tem é R${:.2f}, convertendo para dolar você tem U$
{:.2f}'.format(carteira, dolar))
```

Exercício 11 – Pintando Parede

Faça um programa que leia a largura e a altura de uma parede em metros, calcule a sua área e a quantidade de tinta necessária para pintá-la, sabendo que cada litro de tinta pinta uma área de 2 metros quadrados:

```
Largura da parede: 2.5
Altura da parede: 1.75
Sua parede tem dimensão de 2.5x1.75 e sua área é de 4.375m².
Para pintar essa parede você precisará de 2.1875L de tinta.
```

```
a = float(input('Qual é a altura da parede: '))
l = float(input('Qula é a largura da parede: '))
m = a * l #aqui utilizei uma variável para multiplicar, para saber o metro
quadrado.
tinta = m / 2 #aqui utilizei uma variável para dividir
print('A largura da parede é {}m, e a altura é {}m.'.format(l, a))
print('O metro quadrado da parede é {}M², e vai utilizar {} litro(s) de
tinta.'.format(m, tinta))
```

Exercício 12 – Calculando Descontos

Faça um algoritmo que leia o preço de um produto e mostre seu novo preço, com 5% de desconto:

```
Qual é o valor do Produto? R$123.95
O valor do produto é R$123.95 e com 5% de desconto vai ficar R$117.75.
```

```
produto = float(input('Qual é o valor do Produto? R$'))
desconto = produto - (produto * 5 / 100) #esta formula é para fazer o calculo
de porcentagem e tirar o desconto do valor
print('O valor do produto é R${:.2f} e com 5% de desconto vai ficar R$
```

```
{:.2f}.'.format(produto, desconto))
```

Exercício 13 – Reajuste Salarial

Faça um algoritmo que leia o salário de um funcionário e mostre seu novo salário, com 15% de aumento:

Qual é o salário do funcionário? R$4319.43
O salário do funcionário é R$4319.43 e com 15% de aumento o salario vai ficar R$4967.34.

```
salario = float(input('Qual é o salário do funcionário? R$'))
aumento = salario + (salario * 15 / 100)
print('O salário do funcionário é R${:.2f} e com 15% de aumento '
      'o salario vai ser R${:.2f}.'.format(salario, aumento))
```

Exercício 14 – Conversor de Temperaturas

Escreva um programa que converta uma temperatura digitando em graus Celsius e converta para graus Fahrenheit:

Qual é o valor em graus celsius? 5
O valor é 5.0 C° , convertendo para graus fahrenheit o valor é 41.0 F°.

```
c = float(input('Qual é o valor em graus celsius? '))
f = (c / 5) * 9 + 32 #essa é a formula para fazer a conversao de celsius para fahrenheit.
print('O valor é {}°C , convertendo para graus fahrenheit o valor é {:.1f}°F.'.format(c, f))
```

Exercício 15 – Aluguel de Carros

Escreva um programa que pergunte a quantidade de Km percorridos por um carro alugado e a quantidade de dias pelos quais ele foi alugado. Calcule o preço a pagar, sabendo que o carro custa R$60 por dia e R$0,15 por Km rodado.

Quantos km percorridos o carro fez? Km 720
Por quantos dias o carro foi alugado? 8

São 8 de viagem , e os kms rodados são 720.0 km, tendo um custo de R$588.00 de aluguel.

```
km = float(input('Quantos km percorridos o carro fez? Km '))
dias = int(input('Por quantos dias o carro foi alugado? '))
aluguel = dias * 60.00
kmrodados = km * 0.15
total = aluguel + kmrodados
print('São {} de viagem , e os kms rodados são {} km, '
      'tendo um custo de R${:.2f} de aluguel.'.format(dias, km, total))
```

Aula 8 – Utilizando Módulos – Video aula no site.

Nessa aula vamos aprender como utilizar módulos em Python, utilizando os comando Import e From/import no Python.
Veja como carregar bibliotecas de funções e utilizar vários recursos adicionais nos seus programas utilizando módulos built-in e módulos externos, oferecidos no pypi.

Parte teórica da aula:

Existe uma biblioteca que o seu programa pode necessitar do mesmo jeito que um corpo humano, por exemplo: o corpo humano é apenas um exemplo:

Biblioteca:

Bebida
Água, suco, café, leite e etc…

Comida
Ovo, arroz, sopa, pizza e etc…

Doce
Bolo, pudim, sorvete e etc…

Cada grupo que está aparecendo acima pode ser chamado de bibliotecas, na linguagem de programação, nós podemos fazer importações dessas bibliotecas,os programas em Python tem um conjunto limitado de comandos por padrão, isso funciona para que o program fique leve e com pouca memória, fazendo as funções serem importadas trazendo de fora para os recursos funcionarem perfeitamente. Para incluir algo no Python é necessário usar o comando import:
ex: 1 ° import bebidas
import doces

Para poder importar algo específico de uma biblioteca é necessário colocar, exemplo:

2° from doce import pudim

O 1° exemplo irá importar tudo que estiver na biblioteca e o 2° vai importar somente o que foi mencionado.

Biblioteca padrão do programa Python: math – (significa matemática)

quando você colocar import math, ele vem com uma série de funcionalidades extras, por exemplo: se eu tirei a média de um aluno: 7.25, se eu quiser arredondar o valor para cima tenho que usar a função ceil e se utilizar a função floor vai fazer um arredondamento para baixo.
Função trunc: ele vai eliminar da vírgula para frente sem fazer arredondamento nenhum.
Pow : potência que vai fazer a mesma função do ** potência.
Sqrt : calcular raiz quadrada
factorial: fatorial de um número.
Se eu colocar somente import math ele vai importar tudo o que está acima, se for importar somente uma função tem que utilizar somente esse comando

from math import sqrt – raiz quadrada.

Na biblioteca Random para gerar números aleatórios, para o computador gerar. Ele vai gerar um número real entre 0 e 1, por exemplo: 0.58000, para pedir para aparecer um número inteiro é só digitar randint e limitar os números que você quer que apareça, por exemplo:

```
import random
num = ramdom.randint(1,10)
print(num)
```

Para pode ver as bibliotecas padrão do python e ver exemplos é necessário entras no site Python.org, e para ver os índices de pacotes extras, entrar no mesmo site seção pypi, lá vai ter várias bibliotecas do python que não são padrão.

Exercícios aula 8:

Exercício 16 – Quebrando um número

Crie um programa que leia um número Real qualquer pelo teclado e mostre na tela a sua porção Inteira.

```
Digite um número: 11.025468
A porção inteira do Numero 11.025468 é 11.
```

```
from math import trunc
n = float(input('Digite um número: '))
print('A porção inteira do Numero {} é {}.'.format(n, trunc(n)))
```

Este exercício pode ser feito também sem utilizar a importação de módulos ex:

```
print('A porção inteira do Numero {} é {}.'.format(n, int(n)))
```

Exercício 17 – Catetos e Hipotenusa

Faça um programa que leia o comprimento do cateto oposto e do cateto adjacente de um triângulo retângulo. Calcule e mostre o comprimento da hipotenusa.

```
Digite o comprimento do cateto oposto: 2
Digite o comprimento do cateto adjacente: 2.5
Calculando o 2.0 CO e o 2.5 CA, o resultado é o comprimento da hipotenusa que é 3.20.
```

```
from math import hypot
co = float(input('Digite o comprimento do cateto oposto: '))
ca = float(input('Digite o comprimento so cateto adjacente: '))
h =hypot(ca,co)
#q = (co ** 2) + (ca ** 2)#Essa é a formula para saber a hipotenusa sem a
```

biblioteca
#h = q ** (1/2)
print('Calculando o {} CO e o {} CA, o resultado é o comprimento da hipotenusa que é {:.2f}.'.format(co, ca, h))

e é possível fazer o calculo usando a função sqrt ex: sqrt(co ** 2 + ca ** 2)

Exercício 18 – Seno, Cosseno e Tangente

Faça um programa que leia um ângulo qualquer e mostre na tela o valor do seno, cosseno e tangente desse ângulo:

Digite qualquer ângulo: 30
O ângulo 30.0, tem o seno de 0.50
O ângulo 30.0, tem o cosseno de 0.87
O ângulo 30.0, tem o a tangente de 0.58

```
from math import radians, sin, cos, tan

a = float(input('Digite qualquer ângulo: '))
ar = radians(a)#para calcular o seno, cosseno e tangente com a biblioteca math tem que transformar o ângulo em radianos

print('O ângulo {}, tem o seno de {:.2f}'.format(a, sin(ar)))#formula para saber o seno de radianos.
print('O ângulo {}, tem o cosseno de {:.2f}'.format(a, cos(ar)))#formula para saber o cosseno de radianos.
print('O ângulo {}, tem o a tangente de {:.2f}'.format(a, tan(ar)))#formula para saber a tangente de radianos.
```

Nesse exercício você precisa converter o valor de ângulo para radianos, utilizando a biblioteca math,

Exercício 19 – Sorteando um item na lista

Um professor quer sortear um dos seus quatro alunos para apagar o quadro. Faça um programa que ajude ele, lendo o nome dos alunos e escrevendo na tela o nome do escolhido.

Nome do 1° aluno: ana
Nome do 2° aluno: paulo
Nome do 3° aluno: carlos

Nome do 4° aluno: maria
O nome do aluno escolhido é MARIA

```
from random import choice

a1 = str(input('Nome do 1° aluno: ')).strip().upper()
a2 = str(input('Nome do 2° aluno: ')).strip().upper()
a3 = str(input('Nome do 3° aluno: ')).strip().upper()
a4 = str(input('Nome do 4° aluno: ')).strip().upper()

lista = [a1, a2, a3, a4] #para poder fazer a escolha é necessário fazer uma variante lista, que para o Python fica uma lista que recebe entre colchetes, e dentro desses colchetes fica quem vai participar da lista.

escolha = choice(lista)#
print('O nome do aluno escolhido é {}'.format(escolha))
```

Choice – significa escolha
Para criar uma lista dentro do python você tem que fazer uma variável e colocar entre colchete [] como o exercício acima, variável ‘lista’.

Exercício 20 – Sorteando uma ordem na lista

O mesmo professor do desafio 19 quer sortear a ordem de apresentação de trabalhos dos alunos. Faça um programa que leia o nome dos quatro alunos e mostre a ordem sorteada.

Digite o nome do 1° aluno: Pedro
Digite o nome do 2° aluno: João
Digite o nome do 3° aluno: Ana
Digite o nome do 4° aluno: Maria
A ordem de apreserntação dos alunos é ['PEDRO', 'ANA', 'JOÃO', 'MARIA']

```
from random import shuffle

a1 = str(input('Digite o nome do 1° aluno: ')).strip().upper()
a2 = str(input('Digite o nome do 2° aluno: ')).strip().upper()
a3 = str(input('Digite o nome do 3° aluno: ')).strip().upper()
a4 = str(input('Digite o nome do 4° aluno: ')).strip().upper()

lista = [a1, a2, a3, a4]
```

```
ordem = shuffle(lista)#shuffle significa embaralhar, colocar a lista entre parenteses, mas no print colocar a lista, e assim vai mostrar a lista embaralhado.
print('A ordem de apreserntação dos alunos é {}'.format(lista))
```

Exercício 21 – Tocando um MP3

Faça um programa em Python que abra e reproduza o áudio de um arquivo MP3.

Este exercício tem que fazer o programa tocar uma música.

```
import playsound
playsound.playsound('ex021.mp3')
#O professor fez com pygame eu fiz com a biblioteca playsound, no pygame, para funcionar o som no Linux
#a musica tem que estar em OGG.
```

Aula 9 – Manipulando Texto

Nessa aula, vamos aprender operações com String no Python. As principais operações que vamos aprender são o Fatiamento de String, Análise com len(), count(), find(), transformações com replace(), upper(), lower(), capitalize(), title(), strip(), junção com join().

Manipulando cadeia de texto:

Exemplo: Curso em video no Python

Todo texto no python significa cadeia de texto(Qualquer linguagem de programação entende como uma cadeia de caracteres que se chama strings).
Para o Python toda cadeia de texto está sempre com aspas simples(‘’) ou aspas duplas(“”).
Forma de atribuição de uma string dentro de uma variável. Exemplo abaixo.

Frase = ‘Curso em video em Python’

Quando é feito este tipo de atribuição o python coloca esses dados na memória do computador e ira criar mini espaços e cada letra vai esses mini

espaços, e cada um desses mini espaços vai receber um índice que é um numero sequencial, começando pelo 0 até a ultima letra.
Exemplo:

Esses exemplo abaixo para poder entender é bom ir fazendo no pycharm.

C	u	r	s	o		E	m		P	y	t	h	o	n		V	i	d	e	o
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Isso facilita para poder fazer operações com isso, como a técnica de fatiamento, por exemplo:

Se você colocar entre colchetes []

frase [9] iria aparecer somente o P maiúsculo, mas o python reconhece se a letra é minuscula ou maiúsculo(Isso vai ser explicado a frente) e existe outras maneiras de fatiar.

Por exemplo: frase [9 : 13] - o último número fica de fora.

Se você der print em frase[9:21:2] - ele vai tirar a cada 2 espaços uma letra.

Frase[:5] – se os dois pontos estiver antes do número, ele vai fatiar do inicio até o valor mencionado.

Frase[15:] - Você não sabe o final então vai mostrar a letra do valor mencionado até o final.

Frase[9::3] – nesse caractere, entre os dois pontos não tem um número mostrando o limite final que você quer na letra então ele vai mostrar a letra a partir do 9 ao restante das letras, mas como está constando o 3 ele vai pular as letras e mostrar somente as selecionadas em suas casas de 3 em 3 a partir de 9 ao 20.

Analise: Analizar uma string usando a função (len) que significa comprimento.
Se você usar a funçãp len(frase) – ele vai te mostrar quantos caracteres tem na frase , que no caso do exemplo vai mostrar ‘21’

frase.count (‘o’) – Essa função conta quantas letras tem se repetindo na frase, nesse caso somente vai contar as letras ‘o’ músculas.

Frase.count(‘o’, 0, 13) – Se você colocar dessa forma ele vai contar até a unidade 12 e vai informar que existe somente 1 ‘o’ nesse espaço.

Find – significa encontrar

frase.find(‘deo’) – ele vai te informar em que momento começou, indicando que está na posição 11.

frase.find(‘Android’) – se você utilizar essa função e escrever ‘Android’ sendo que não existe essa frase vai constar -1 e com isso quer dizer que essa frase não foi encontrado (não existe).

‘Curso’ in frase – existe a palavra curso na frase, o in significa se existe o item em frase, e vai mostra True ou False.

Transformação :

replace() – significa trocar, reposicionar.

Uma lista de string é imutável, nós não conseguimos mexer nela, mas você consegue mudar ela através dos métodos.

Frase.replace(‘Python’, ‘Android’) – onde estiver escrito ‘Python ele vai substituir por ‘Android’. O replace substitui a palavra de uma forma secundária.

Upper – significa para cima ou maiúscula.
Frase.upper() - mantém o que está em maiúscula e transforma o restante em maiúscula.

Frase.lower() - mantém as letras em minúscula e transforma as maiúsculas em minúsculas.

Essas funções é necessário sempre colocar os parênteses no final.

Capitalize – Deixar a string capitalizado.
Frase.capitalize – ele vai joga tudo para minúscule vai manter somente a 1° letra em maiúscula.

Frase.title() - a função title, ele analisa quantas palavras tem a função string e vai deixar todas as palavras com as 1° letras em maiúscula. , exemplo:

Curso em Video Python

Strip()- remove todos os espaços inúteis no inicio e fim.

De forma similar ao strip que é o rstrip()
r – de right que significa á direita.
Atenção .Muitas funções dentro do Python que tratam as strings tem a função r ou l

lstrip() - o l é de left que significa esquerda
frase.lstrip() - ele retira os espaços da esquerda e mantem os da direita.

Funções de Divisão – poder dividir strings

frase.split() - ele vai dividir uma strings em uma lista onde cada elemento vai ser separado pelo seu caracteres de split que por padrão é um espaço.

Junção – se eu tenho nomes separados em listas, você consegue juntar uma coisa na outra usando o join

‘-’.join(frase) – Você vai juntar todos os elementos da frase e vai juntar e vai utilizar o -, se quiser tirar o – e só colocar espaços.

Para colocar um texto em print, tem que colocar ‘’’ no print na hora que for escrever.

Exercício 22 – Analisador de Textos

Crie um programa que leia o nome completo de uma pessoa e mostre:
– O nome com todas as letras maiúsculas e minúsculas.
– Quantas letras ao todo (sem considerar espaços).
– Quantas letras tem o primeiro nome.

Seu nome em letra maiúscula fica (PAULO DE SOUSA MARQUES).
Seu nome com letra minúscula fica (paulo de sousa marques).
O seu nome completo tem 19 letras.
O seu Primeiro nome tem 5 letras.

```
nome = str(input('Digite seu nome inteiro: '))
```

```
print('Seu nome em letra maiúscula fica ({}).'.format(nome.upper()))

print('Seu nome com letra minúscula fica ({}).'.format(nome.lower()))

#lista = nome.split()

print('O seu nome completo  tem {} letras.'.format(len(nome) - nome.count(' ')))

#print('O seu nome completo tem {} letras'.format(len(''.join(lista))))

print('O seu Primeiro nome tem {} letras.'.format(nome.find(' ')))

#print('O seu primeiro nome tem {} letras.'.format(len(lista[0])))
```

#A FORMA QUE ESTÁ APAGADO É O MODO QUE O PROFESSOR FEZ E O QUE ESTÁ ATIVO É DA FORMA QUE FIZ.
Este exercício você consegue resolver com a função len, e join também.

Exercício 23 – Separando dígitos de um número:

Faça um programa que leia um número de 0 a 9999 e mostre na tela cada um dos dígitos separados.

```
Informe um número: 3224
Analisando o Número: 3224.
A unidade é 4
A dezena é 2.
A centena é 2.
O milhar é 3.

Informe um número: 23
Analisando o Número: 23.
A unidade é 3
A dezena é 2.
A centena é 0.
O milhar é 0.
```

```
n = int(input('Informe um número: '))
print('Analizando o Número: {}.'.format(n))

n1 = n // 1 % 10
n2 = n // 10 % 10
```

```
n3 = n // 100 % 10
n4 = n // 1000 % 10

print('A unidade é {}'.format(n1))
print('A dezena é {}.'.format(n2))
print('A centena é {}.'.format(n3))
print('O milhar é {}.'.format(n4))
```

Para converter uma variante basta colocar como o exemplo :
num = str(n) – para converter um número inteiro para uma string.
Isso no caso se a variante tiver em int e você querer converter para string.

É possível fazer esse exercício utilizando estrutura de repetição que vai ser ensinado no mundo 2.

Exercício 24 – Verificando as primeiras letras de um texto:

Crie um programa que leia o nome de uma cidade diga se ela começa ou não com o nome “SANTO”.

```
Digite o nome de uma cidade: Santo antonio
O nome da sua cidade é SANTO ANTONIO e é True que começa com "SANTO"

Digite o nome de uma cidade: Campo LImpo Paulista
O nome da sua cidade é CAMPO LIMPO PAULISTA e é False que começa com "SANTO"
```

```
cidade = str(input('Digite o nome de uma cidade: ')).strip().upper()
print('O nome da sua cidade é {} e é {} que começa com "SANTO"'.format(cidade, cidade[:5] == 'SANTO'))
```

Quando você coloca os sinais de == ‘palavra’ como acima, você está pedindo para verificar se é ou não, então vai informar se é True ou False.
Para não ter problemas transformar a palavra para maiúscula ou minúscula, para na hora que você for digitar não gerar nenhum erro.

Exercício 25 – Procurando uma string dentro de outra

Crie um programa que leia o nome de uma pessoa e diga se ela tem “SILVA” no nome:

Digite seu nome em extenso: David Edison da Rocha
O seu nome é DAVID EDISON DA ROCHA, e é False que tem "SILVA" em seu nome.

Digite seu nome em extenso: Paulo silva Pereira
O seu nome é PAULO SILVA PEREIRA, e é True que tem "SILVA" em seu nome.

```
nome = str(input('Digite seu nome em extenso: ')).strip().upper()
print('O seu nome é {}, e é {} que tem "SILVA" em seu nome.'.format(nome, 'SILVA' in nome))
```

Exercício 26 – Primeira e última ocorrência de uma string:

Faça um programa que leia uma frase pelo teclado e mostre quantas vezes aparece a letra “A”, em que posição ela aparece a primeira vez e em que posição ela aparece a última vez:

Digite uma frase: Amanda ama o pedro
O sua frase é AMANDA AMA O PEDRO, e aparece 5 vezes a letra "A".
A primeira letra "A" esta na posição 1.
A ultima letra "A" está na posição 10.

```
frase = str(input('Digite uma frase: ')).strip().upper()
print('O sua frase é {}, e aparece {} vezes a letra "A".'.format(frase, frase.count('A')))
print('A primeira letra "A" esta na posição {}.'.format(frase.find('A')+1))
print('A ultima letra "A" está na posição {}.'.format(frase.rfind('A')+1))
```

para ver qual é a posição da ultima letra é só colocar rfind() + 1 como está acima

Exercício 27 – Primeiro e último nome de uma pessoa

Faça um programa que leia o nome completo de uma pessoa, mostrando em seguida o primeiro e o último nome separadamente:

Digite seu nome completo: David Edison da Rocha

Seu nome completo é DAVID EDISON DA ROCHA
O seu primeiro nome é DAVID.
O seu ultimo nome é ROCHA.

```
nome = str(input('Digite seu nome completo: ')).strip().upper()
lista = nome.split()
print('Seu nome completo é {}'.format(nome))
print('O seu primeiro nome é {}.'.format(lista[0]))
print('O seu ultimo nome é {}.'.format(lista[-1]))
#se voce utilizar lista[len(lista)-1], faz a mesma coisa.
```

Aula 10 – Condições em Python if.else

Condições (Parte 1)

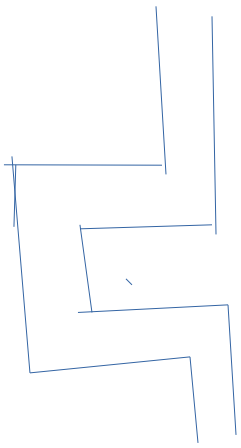
Nessa aula vamos aprender a utilizar estruturas condicionais simples e compostas nos seus programas em python.

Teoria:

Exemplo:
Existe um ponto de chegada e um ponto de destino, ai você se perguntaria, como você iria para sair do ponto de chegada até ao ponto de destino?

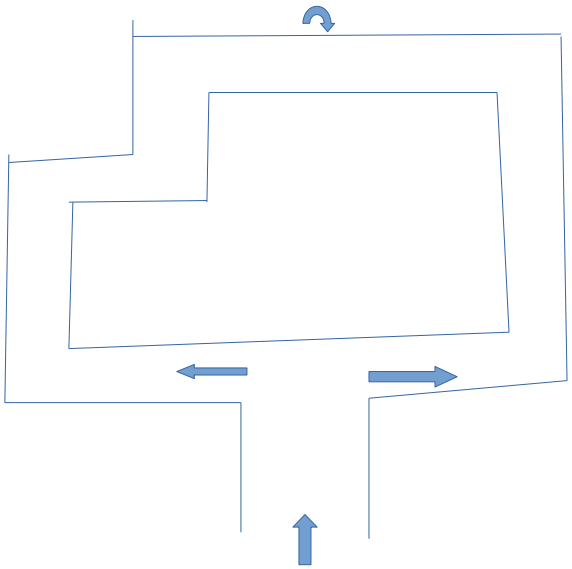
Carro – é o objeto / Siga – é o método, e sempre tem parênteses no final

Carro.siga( )
Carro.esquerda( )
Carro.siga( )
Carro.direita( )
Carro.siga( )
Carro.direita( )
Carro.siga( )
Carro.pare( )



Todos os comandos são sequenciais e começa de cima para baixo, você não pode pular etapas, se não vai dar erro e esse exemplo só existe um caminho, utilizando uma estrutura sequencial.

Se essa sequência for modificado, provavelmente não vai chegar no destino final, o passo a passo vai estar errado.
Nem sempre os caminhos é simples ou facil de ser executados igual ao exemplo abaixo:



Carro.siga( )

No exemplo a cima após o ponto siga você terá duas possibilidades, como ir para a esquerda ou ir para a direita.
Se você escolher um caminho você não vai fazer o outro, com isso você tem a possibilidade dentro do programa, nesse caso o programa deixa de ser executado de cima para baixo todas as linhas, com isso você consegue fazer desvios tendo a possibilidade dentro do programa.
De acordo com a decisão de uma pessoa, ela vai fazer o caminho da direita ou da esquerda.
Se você virar para a esquerda o algoritimo e o subalgoritmo vai ficar da seguinte forma:

Carro.Siga.( ) - Algorítimo que sempre vai acontecer.
Carro.Siga( )
Carro.Direita( )
Carro.Siga( )
Carro.Direita( ) Subalgoritimo – esse só vai acontecer se você virar
Carro.Esquerda( ) para a esquerda
Carro.Siga( )

```
    Carro.Direita( )
    Carro.Siga( )
Carro.Pare( )  - Algorítimo fixo
```

Se você virar para a direita vai fazer da maneira abaixo:

```
Carro.siga( )
    Carro.siga( )
    Carro.esquerda( )
    Carro.siga( )
    Carro.esquerda( )
    Carro.siga( )
Carro.pare( )
```

Os dois caminhos sõa diferentes nas quantidades de comandos de um para o outro, os comandos da esquerda é maior que os comandos da direita.
Lembrando: Todo programa existe as suas condições, e o exemplo acima é uma forma de explicar isso.

Exemplo:

```
                                    Carro.siga( )

    Carro.siga( )                                   Carro.siga( )
    Carro.Direita( )                                Carro.esquerda( )
    Carro.Siga( )                                   Carro.siga( )
    Carro.Direita( )                                Carro.esquerda( )
    Carro.Esquerda( )                               Carro.siga( )
    Carro.Siga( )
    Carro.Direita( )
    Carro.Siga( )

                                    Carro.pare( )
```

Independente do caminho que você escolher o 1° e o último comando sempre irão acontecer, os outros comandos irão acontecer somente se vocÊ decidir virar para a esquerda ou para a direita.
Com isso no mesmo programa você vai criar dois programas diferentes.

E nesse caso você tem dois fluxo, o primeiro vai ser chamado de fluxo verde e o outro vai ser chamado de fluxo vermelho.
Cada um deles vai chegar ao mesmo destino, a decisão vai ser feita de acordo com o andamento do programa.
Para a programação vai ser utilizada a estrutura que se chama se.

Carro.siga( )

se Carro.esquerda( ) se não

Carro.siga( ) Carro.siga( )
Carro.direita( )… Carro.esquerda( )…

Se o carro for para a esquerda ele se segue o caminho até pare, se não ele vai fazer o caminho da direita até o pare, mas no python é necessário fazer uma representação estruturada ou representação identada.

```
Carro.Siga.( )
se carro.esquerda( ):
    Carro.Siga( )
    Carro.Direita( )
    Carro.Siga( )
    Carro.Direita( )
    Carro.Esquerda( )
    Carro.Siga( )
    Carro.Direita( )
    Carro.Siga( )
```

Tem identação(espaço para dentro)

Se não:

```
    Carro.siga( )
    Carro.esquerda( )
    Carro.siga( )
    Carro.esquerda( )
    Carro.siga( )
Carro.pare( )
```

Essa é a forma para colocar no python e no exemplo acima o se está para fora e o conjunto está um pouco para dentro, e repete o mesmo com o se não finalizando com o pare para fora.
Esse espaço se chama identação, os que não tem identação, significa que eles irão acontecer todas as vezes que o programa for executado, no caso o caminho siga e o pare irão sempre acontecer.
A estrutura condicional é tratada de maneira abaixo:

```
se carro.esquerda( )
    Bloco-v-
se não
    Bloco-e-
```

O espaço para dentro você vai utilizar a tecla Tab do teclado para o python, e a forma dos comando tem que ser em inglês:
if carro.esquerda( ) no lugar do se você coloca if e no final da linha você coloca dois pontos.
No lugar do se não você coloca eles e dois pontos.

```
If carro.esquerda( ):
        Bloco True
else:
        Bloco False
```

O exemplo acima é sua primeira representação de uma estrutura condicional, e em uma condição, ou o bloco verdadeiro é executado ou o bloco falso, nunca serão os dois programas ao mesmo tempo.
Abaixo tem um exemplo se o seu carro é velho ou novo:

```
tempo = int(input('Quantos anos tem o seu carro? '))
if tempo < = 3:
        print('Carro novo')
else:
        print('Carro velho')
print('—Fim--')
```

Todo comando que estiver colado do lado esquerdo da tela vai ser executado sempre.

Todo comando que estiver com identação pode ser executado ou não, Vai depender da resposta do usuário, são dois programas, são dois fluxos, são dois algoritimos em um proama só.
É possível fazer esse programa com apenas 3 linhas que é uma condição simplificada.:

```
tempo = int(input('Quantos anos tem seu carro? '))
print('Carro novo' if tempo < = 3 else'Carro Velho')
print('—Fim--')
```

se tiver no programa somente o if ele é uma condição simples, se estiver o if e o else é uma condição composta e a condição simpificada é tudo em um print só.ex:

```
print('Parabéns' if >=6 else 'Estude mais')
```

Atenção:
Ao utilizar o and e o or nos programas, o and vai estar junto com o if e o or é para escolher outro.

Exercício 28 – Jogo da Adivinhação v.1.0

Escreva um programa que faça o computador "pensar" em um número inteiro entre 0 e 5 e peça para o usuário tentar descobrir qual foi o número escolhido pelo computador. O programa deverá escrever na tela se o usuário venceu ou perdeu.

```
from random import randint
from time import sleep #biblioteca time, para colocar tempo no seu programa.

n = int(input('Adivinhe qual número o computador escolheu. entre 0 a 5:
'))#aqui o jogador tenta adivinhar qual número é.

número = randint(0,5)#Faz o computador pensar.

print('Processando …')

sleep(3)#sleep é um método que faz o computador pausar por alguns segundos, os segundo é estipulado pelo programador.
```

```
print('O seu número é {} e o computador escolheu {}.'.format(n, número))

if n == número: #Aqui o jogador acertou.
    print('Parabéns! você adivinhou o número do computador, você venceu !!!')
else:#Aqui o jogador errou.
    print('Que pena, não é o mesmo número, você perdeu !!!')
```

```
Adivinhe qual número o computador escolheu. entre 0 a 5: 1
Processando ...
O seu número é 1 e o computador escolheu 3.
Que pena, não é o mesmo número, você perdeu !!!
```

ou

```
Adivinhe qual número o computador escolheu. entre 0 a 5: 4
Processando ...
O seu número é 4 e o computador escolheu 4.
Parabéns! você adivinhou o número do computador, você venceu !!!
```

Exercício 29 – Radar eletrônico

Escreva um programa que leia a velocidade de um carro. Se ele ultrapassar 80Km/h, mostre uma mensagem dizendo que ele foi multado. A multa vai custar R$7,00 por cada Km acima do limite.

```
Qual é a velocidade do carro? KM 120
A velocidade do carro é 120.0 km/h.
Você foi multado! Você ultrapassou 40.0km/h.
Sua Multa vai ser do valor de R$280.00.
```

Ou

```
Qual é a velocidade do carro? KM 80
A velocidade do carro é 80.0 km/h.
Parabéns! Continue dirigindo com segurança !
```

```
carro = float(input('Qual é a velocidade do carro? KM '))
print('A velocidade do carro é {} km/h.'.format(carro))
if carro > 80:
```

```
    velocidade = carro - 80
    multa = velocidade * 7.00
    print('Você foi multado! Você ultrapassou {}km/h. \nSua Multa vai ser do
valor de R${:.2f}.'
    .format(velocidade, multa))
else:
    print('Parabéns! Continue dirigindo com segurança !')
```

Este execício pode ser feito sem o else, tirando o else e utilizando somente o if, é uma condição simples.
#nesse caso você vai colocar um print mostrando 'Tenha uma boa viagem, Dirija coom segurança' e assim vai aparecer isso para ambos.
#e não vai precisar do else, e nem vai desejar parabéns.

Exercício 30 – Par ou Ímpar?

Crie um programa que leia um número inteiro e mostre na tela se ele é PAR ou ÍMPAR.

Digite um número: 2
O seu número é 2
Esse número é Par.

Ou

Digite um número: 3
O seu número é 3
Esse numero é Impar.

```
n = int(input('Digite um número: '))

int = n % 2 #O resto da divisão por 2 de qualquer número par é 0, e de
qualquer número ímpar é 1

print('O seu número é {}'.format(n))
if int == 0:
    print('Esse número é Par.')
else:
    print('Esse numero é Impar.')
```

Exercício 31 – Custo da Viagem

Desenvolva um programa que pergunte a distância de uma viagem em Km. Calcule o preço da passagem, cobrando R$0,50 por Km para viagens de até 200Km e R$0,45 para viagens mais longas.

Quantos km é a sua viagem? Km 200
Sua Viagem é de 200.0km, e vai ter um custo de R$100.00

Ou

Quantos km é a sua viagem? Km 250
Sua Viagem é de 250.0km, e vai ter um custo de R$112,50

```
v = float(input('Quantos km é a sua viagem? Km '))

if v <= 200:#se o valor for menor ou igual a 200 a resposta vai ser esta.
    valor = v * 0.50#Calculo para saber o custo da viagem
else:# se for acima de 200 vai ser esta.
    valor = v * 0.45 #CAlculo para saber o custo da Viagem.
print('Sua Viagem é de {}km, e vai ter um custo de R${:.2f}'.format(v, valor))

#neste exercício você pode colocar com uma condição simplificada também.
#ex: valor = v * 0.50 if v <= 200 else valor v * 0.45
```

Exercício 32 – Ano Bissexto

Faça um programa que leia um ano qualquer e mostre se ele é bissexto.

Veja qual ano é Bissexto e qual não é !
Digite um ano qualquer, se quiser mostrar o ano atual digite 0: 2016
O ano que você digitou é 2016.
O ano é Bissexto

Ou

Veja qual ano é Bissexto e qual não é !
Digite um ano qualquer, se quiser mostrar o ano atual digite 0: 2021
O ano que você digitou é 2021.
Este ano não é Bissexto !

Ou

Veja qual ano é Bissexto e qual não é !
Digite um ano qualquer, se quiser mostrar o ano atual digite 0: 0
O ano que você digitou é 2021.
Este ano não é Bissexto !

```
from datetime import date#Bibliote e função para saber a data

print('Veja qual ano é Bissexto e qual não é !')
data = int(input('Digite um ano qualquer, se quiser mostrar o ano atual digite 0: '))

if data == 0:
    data = date.today().year#função para saber a data atual

print('O ano que você digitou é {}.'.format(data))

if data % 4 == 0 and data % 100 != 0 or data % 400 == 0:#formula para saber se o ano é bissexto
    print('O ano é Bissexto')
```

Exercício 33 – Maior e menor valores

Faça um programa que leia três números e mostre qual é o maior e qual é o menor.

Digite o 1° número: 7
Digite o 2° número: 2
Digite o 3° número: 4
O maior número é 7.
O menor número é 2.

```
n1 = int(input('Digite o 1° número: '))
n2 = int(input('Digite o 2° número: '))
n3 = int(input('Digite o 3° número: '))
```

```
if n1 > n2 and n1 > n3:
    maior = n1
if n2 > n1 and n2 > n3:
    maior = n2
if n3 > n1 and n3 > n2:
    maior = n3
if n1 < n2 and n1 < n3:
    menor = n1
if n2 < n1 and n2 < n3:
    menor = n2
if n3 < n1 and n3 < n2:
    menor = n3

print('O maior número é {}.'.format(maior))
print('O menor número é {}.'.format(menor))

#Você pode retirar um if já considerando o n1 menor e começando o if a partir do n2 o mesmo acontece para o maior.
```

Exercício 34 – Aumentos múltiplos

Escreva um programa que pergunte o salário de um funcionário e calcule o valor do seu aumento. Para salários superiores a R$1250,00, calcule um aumento de 10%. Para os inferiores ou iguais, o aumento é de 15%.

Digite o salário do funcionário: R$9000
O salário do funcionário é R$9000.00, e com 10% de aumento fica R$9900.00.

Ou

Digite o salário do funcionário: R$1100
O salário do funcionário é R$1100.00, e com 15% de aumento fica R$1265.00.

```
salario = float(input('Digite o salário do funcionário: '))

if salario <= 1250.00:
    valor = 15#Coloquei essa variável para mostrar a porcentagem
```

```
    aumento = salario + (salario / 100 * 15)#Variável para calculo do aumento de salário.

else:
    valor = 10
    aumento = salario + (salario / 100 * 10)

print('O salário do funcionário é R${:.2f}, e com {}% de aumento fica R${:.2f}.'
      .format(salario, valor, aumento))
```

Exercício 35 – Analisando Triângulo v1.0

Desenvolva um programa que leia o comprimento de três retas e diga ao usuário se elas podem ou não formar um triângulo.

```
Analisador de triângulos:
Digite a 1° reta: 9
Digite a 2° reta: 7
Digite a 3° reta: 8
Elas podem formar um triângulo!
```

Ou

```
Analisador de triângulos:
Digite a 1° reta: 3
Digite a 2° reta: 2
Digite a 3° reta: 10
Elas não podem formar um triângulo!
```

```
print('Analisador de triângulos: ')
reta1 = float(input('Digite a 1° reta: '))
reta2 = float(input('Digite a 2° reta: '))
reta3 = float(input('Digite a 3° reta: '))

if reta1 < reta2 + reta3 and reta2 < reta3 + reta1 and reta3 < reta1 +
reta2:#Formula para saber se três retas
#formam um triângulo
```

```
    print('Elas podem formar um triângulo!')
else:
    print('Elas não podem formar um triângulo!')
```

Aula 11 – Cores no Terminal

Nessa aula vamos aprender como utilizar os códigos de escape sequence Ansi para configurar cores para os seus programas em python, veja como utiliza o código \033[m com todas as suas principais possibilidades.
Esta aula é extra, porque não é tão importante para os programas, a única coisa que pode ser feito é deixar os programas bonitos colocando cores.
Padrão Ansi Escape sequecis é um padrão de normalização internacional e o scape sequencis funciona em vários ambientes, tudo dentro de Ansis começa com contra barrae um código.
Para poder adicionar cores no python é necessário colocar \033[m fechando o código com a letra m.
O código 033 é o código que funciona melhor para o python, entre o colchete [e o m, você vai colocar o código das cores.
1° código é o código de comportamento que é o - style
2° Código é o do texto – Text
3° código é de fundo – Background

Esse códigos você pode colocar em qualquer ordem, mas você pode preencher o \033[m com 1 código, com 2 ou 3 códigos ou nenhum códigos.
O exemplo abaixo é adicionando os 3 códigos.
Exemplo:

\033[0; 33; 44m

Todo código ansi para cores no python começa como o exemplo acima. E os três códigos acima são opcionais, pode ser colocado os três códigos ou um ou dois códigos e pode colocar na ordem que quiser finalizando com a letra m.

Código de style que funciona melhor no python:
0, 1, 4, 7
0 – none, nenhum – sem estilo
1 – Bold – negrito
4 – underline – sublinhado
7 – negative – vai inverter as configurações

Códicos de cores de text para o python:

30 – Branco
31 – Vermelho
32 – Verde
33 – Amarelo
34 – Azul
35 – Roxo
36- Verde mar
37- Cinza
o 37 que é cinza é a cor padrão.
Para adicionar mais cores é necessário adicionar uma biblioteca.
Códigos de cores do Background – cores de fundo.

40 – Branco
41 – Vermelho
42 – Verde
43 – Amarelo
44 – Azul
45 – Roxo
46- Verde mar
47- Cinza
aqui não existe um código para a cor preta, mas podemos gerar o preto.

Teste – Fundo vermelho letra branca:
\033[0;30;41m
Teste – Fundo azul, letra amarela:
\033[4;33;44m
Teste – Fundo amarelo, letra negrito roxo:
033[1;35;43m

Teste – Fundo verde, letra branca, sem o códico de style, vai utilizar a forma padra do python:
\033[30;42m
Teste – Letra preta, fundo preto é a forma padrã do python, não precisa colocar código nenhum, somente o código ansi:
\033[m
Teste - Fundo branco, texto branco, vai utilizar o código 7, o de inversão:
\033[7;30m trinta é branco, mas invertendo fica preto.
Este inversor pode ser usadas com todas as cores invertendo elas.
Obs: Lembrando que os códigos tem que ser separados po ; ponto e virgula.

Prática:
Para colocar cor nas letras tem que ser de dentro do print ou palavras que estão dentro das aspas, exemplo abaixo:

```
print('\033[1;31;40m Olá Mundo! \033[m')
```

se não cplocar o \033[m no final a cor vai até o final da linha se for print, se for uma variante vai até o ponto de resposta, colocando \033[m no final, você está tirando a formatação.
Existe uma biblioteca chamada colorise.
É possível colocar cores no format, exemplo:

```
nome = 'Guanabara'
print('Olá! Muito prazer em te conhecer, {} {} {}'.format('033[4,34m', nome, '\
033[m'))
```

Tem que colocar a quantidade de chaves que foi colocado os códigos de cores.
Pode ser feita também cores no sistemas como uma variante.

```
Nome = 'Guanabara'
cores = {'limpa' : '\033[m', 'Azul':'\033[34m', 'Amarelo': '\033[33m', 'Preto e Branco' : '\033[7;30m'}
print('Olá! Muito Prazer em te conhecer, {}{}{}!!!'.format(cores['preto e branco'], nome, cores['limpar']))
```

Para cores em vez de parênteses tem que colocar chavez e é uma forma organizada de deixar seu programa mais simples.