transforme-se

serasa experian

WISEHANDS
Tecnologia e Acessibilidade

Na aula de hoje

- Correção de exercícios
- Funções
- Parâmetros
- Retorno
- Chaves
- *args e **kwargs

Correção do Exercício

Utilizando o for no jogo e novas funcionalidades

Para casa. Entrega código no github até às 19h do dia 26/06.
Valendo nota de participação. Grupos de 1 a 6 pessoas, escolha livre.

- Substitua o comando while pelo comando for.
- Escolha um número secreto entre 0 e 100.
- Crie um nível de dificuldade para o jogo.
  Peça para o usuário escolher em qual nível ele deseja jogar. O nível é mensurado de acordo com as tentativas.
- Acrescente um sistema de pontuação.
  O jogador deve começar com 1000 pontos e cada chute errado deve ser subtraído do total de pontos um valor que corresponde a diferença entre o chute e o número secreto.
  (dica, pesquise sobre a função abs(), ela pode ser útil)

Correção Atividade

Entrega junto com o jogo da adivinhação

Dada a lista = [13, -3, 5, 9, 19, 46, 79, 37, -18, 3, 13, 7, 4, 4, -42], faça um programa que:

- Imprima o maior elemento;
- Imprima o menor elemento;
- imprima os números pares;
- imprima o número de ocorrências do primeiro elemento da lista;
- imprima a média dos elementos;
- imprima a soma dos elementos de valor negativo.

Funções

É parecido com o conceito da matemática

Em Python, uma função é uma sequência de comandos que executa alguma tarefa e que tem um nome.

A sua principal finalidade é nos ajudar a organizar programas em pedaços que correspondam a como imaginamos uma solução do problema.

A sintaxe de uma definição de função é:

```
def NOME( PARÂMETROS ):
    COMANDOS
```

Funções

Pensando em *matematiquês*

Exemplo, calcular x:

f(x) = 2x + 5

f(1) = ?, f(2) = ?

Exemplo, calcular a razão do espaço pelo tempo:

f(espaço, tempo) = espaço / tempo

velocidade(espaço, tempo) = espaço / tempo

velocidade(100, 20) = ?

Funções

Agora, em python

Exemplo, calcular x:

```
def funcao(x):
    resultado = 2*x + 5
    print("Para x = {} o valor da função é {}".format(x, resultado))
```

Exemplo, calcular a razão do espaço pelo tempo:

```
def velocidade(espaco, tempo)
    v = espaco/tempo
    print("velocidade: {} m/s".format(v))
```

Parâmetro e Argumento

Parâmetro muitas vezes é utilizado como sinónimo de argumento

O termo parâmetro muitas vezes é utilizado como sinónimo de argumento, mas geralmente **utiliza-se "parâmetros" quando se faz referência às variáveis situadas entre os parênteses** de um método ou função e **"argumentos" são os valores atribuídos a esses parâmetros**

Um **conjunto de parâmetros consiste em uma lista com nenhum ou mais elementos** obrigatórios ou opcionais.

Para um parametro ser opcional, é atribuido um valor padrão, normalmente o none.

Parâmetro e Argumento

Exemplo:

```
def dados(nome, idade=None):
    print('nome: {}'.format(nome)
    if(idade is not None):
        print('idade:{}'.format(idade)
    else:
        print('idade nao informada')
```

Retorno

E se quisermos apenas que retorne um valor?

Para calcular a aceleração, é preciso apenas do valor numérico da velocidade:

aceleração = velocidade(parâmetros) / tempo

Mudando a função velocidade:

```
def velocidade(espaco, tempo):
    v = espaco / tempo
    return v
```

Agora é possível calcular a aceleração

Retorno

Uma função pode ter mais comando de um retorno.

```
def dados(nome, idade=None):
    print('nome: {}'.format(nome)
    if(idade is not None):
        return('nome: {} \n idade:{}'.format(nome,idade))
    else:
        return('nome: {} \n idade: não informada'.format(nome))
```

Mesmo tendo mais de um comando de retorno, ela executa apenas um.

Quando a função encontra um return, ela para de executar a função.

*ARGS

Número arbitrário de parâmetros

O *args é usado quando não sabemos quantos argumentos vamos passar para uma função:

```
def teste(arg, *args):
    print('o primeiro argumento é normal: {}'.format(arg)
    for arg in args:
        print('os outros argumentos: {}'.format(arg)
```

**KWARGS

Número arbitrário de chaves

O **KWARGS permite passar o tamanho variável da palavra chave dos argumentos para uma função:

```
def funcao:(**kwargs)
    for key, value in kwargs.item():
        print("{0} = {1}".format(key, value))
```

O ***args espera uma tupla** de elementos posicionais, o ****kwargs espera um dicionário** com argumentos nomeados

Exercicios em sala

- Vamos, juntos criar uma função para testar e brincar com ***args** e com ****kwargs**

- Vamos criar um menu de jogos.
  Importar **o jogo da forca** e **o jogo da adivinhação** e o **usuário escolhe** qual quer jogar.

Atividade para casa

- Crie uma função calculadora() que receba dois números
  e retorne o resultado das 4 operações básicas da matemática entre eles.

- Crie uma função divisão(), que receba dois números como parâmetros
  e retorne o resultado da divisão do primeiro pelo segundo.

- Modifique a função feita em sala velocidade() para que utilize a função divisão
  para calcular a velocidade.
- Para ser entregue no dia 03/07 às 19h. Grupos entre 1 e 6 pessoas

transforme-se

O conhecimento é o poder
de transformar o seu futuro.