Curso de Computação 1
Introdução à Programação em Python

AULA 3

1. Tipos de Dados

Na primeira parte da aula, vamos estudar sobre os tipos de dados básicos do Python. Já falamos na aula anterior sobre os tipos de dados numéricos (int, float e complex). Mas na computação atual fazemos muito mais do que apenas processamentos numéricos. Editores de texto e aplicativos de redes sociais, por exemplo, são muito mais do que calculadoras. Os programas que estão por trás dessas aplicações precisam então saber lidar com dados não numéricos também. Que outros tipos de dados podemos encontrar em uma linguagem de programação? Quais são as características de cada um? Que tipo de processamento pode ser realizado com dados desses diferentes tipos? Sabemos realizar operações aritméticas em dados dos tipos numéricos, por exemplo. Mas quanto aos outros tipos de dados? Podemos transformar dados de um tipo em outro? Para responder essas perguntas, veja o vídeo abaixo:

- Tipos de Dados

Dados numéricos (inteiro, ponto flutuante e complexo)

A melhor maneira de entender como o Python lida com os tipos de dados é vendo como se comportam as operações pré definidas do Python para estes tipos. Lembrando que os tipos de dados numéricos são: inteiro (int), ponto flutuante (float) e complexo (complex).

Exercício: Vamos verificar agora no shell IDLE os comportamentos que foram apresentados no vídeo. Abra o IDLE e digite no prompt (>>>) do shell as seguintes expressões numéricas e observe o resultado da avaliação de cada expressão.

Subtração, soma ou multiplicação entre inteiro e float:
>>> 595 - 2.0

>>> 500 * 2. (atenção! Há um ponto no final da expressão)

>>> 3e2 + 100

Subtração, soma ou multiplicação entre inteiro ou float e complexo. Observe que os componentes reais e imaginários do número complexo só vão ser mostrados com o ponto decimal caso seja necessário (caso a parte fracionária seja diferente de zero).

```
>>> 595 - (1 + 2j)

>>> 595. - (1 + 0j)

>>> 2 * (1.5 + j)

>>> 2 * (1.5 + 1j)
```

E as divisões? Como elas se comportam em relação aos diferentes tipos?

```
>>> 2 / 3

>>> 2 // 3

>>> 2 // 3.0
```

Você lembra como é a divisão de números complexos? Se não lembra, faça uma pesquisa na internet para relembrar, senão você não terá como saber se o interpretador Python está fazendo a conta que você deseja que seja feita.

```
>>> (2+2j)/4

>>> 2.5/(1+1j)
```

Verifique também o resultado da avaliação de expressões contendo conversões explícitas de tipos, sempre digitando a expressão no prompt do shell. Algumas podem não ser possíveis em Python! Descubra quais são dentre as abaixo:

```
>>> float(3)

>>> complex(3.1)

>>> int(3.1)

>>> int(2.5+2j)

>>> float(2.5+2j)
```

Dados textuais (strings)

Agora que já experimentamos bastante com expressões numéricas, vamos exercitar expressões sobre strings.

Exercício: Continue no shell do IDLE, digite as seguintes expressões e observe os resultados. Observe que algumas das sequências de caracteres a seguir contém caracteres “em branco”, ou seja, espaços, e algumas das expressões propostas podem não ser válidas.

```
>>> 'primeira' + 'segunda'

>>> “primeira,” + ' segunda e ' + terceira

>>> “primeira,” + ' segunda e ' + “terceira.”

>>> 5 * ’a’

>>> ‘a’ * 5.0

>>> 5 + ‘a’

>>> str(5)
```

Atividade: Chegou a hora de testar os seus conhecimentos. Após terminar este exercício, responda às perguntas da atividade “tipos de dados”. Fique atento ao prazo de entrega dessa atividade!

2. Operadores e Expressões Booleanas

Na segunda parte da aula, veremos como comparar valores de dados, através de *expressões booleanas,* ou seja, expressões cujo valor, uma vez calculado, é do tipo de dados booleano (bool em Python). Estamos acostumados a usar em matemática operadores de comparação (igual, diferente, maior que, etc.). Esses operadores são muitas vezes a base das expressões booleanas. Mas temos também os chamados *operadores booleanos*, que veremos agora. Para utilizar expressões booleanas, é preciso compreender bem como funciona cada operador booleano. Veja no vídeo a seguir o que são expressões booleanas, quais são os operadores booleanos, para que servem e qual a ordem de precedência desses operadores em uma expressão:

- Operadores e Expressões Booleanas

No vídeo a seguir, você verá vários exemplos de problemas resolvidos a partir da criação de funções utilizando expressões booleanas, e poderá também compreender na prática o que faz com que uma expressão possa ser considerada booleana ou não.

- Exemplos: Operadores e Expressões Booleanas

Atividade: Chegou a hora de testar os seus conhecimentos. Após terminar este exercício, responda às perguntas da atividade “Expressões Booleanas”. Fique atento ao prazo de entrega dessa atividade!

3. Estrutura Condicional

Na terceira e última parte da aula, vamos estudar a formação de Estruturas Condicionais em Python. Em diversas situações, precisamos desenvolver códigos que possuam partes que sejam executadas ou não, de acordo com condições específicas. Através do uso de estruturas condicionais, podemos satisfazer esse requisito. Veja no vídeo a seguir como são formadas as estruturas condicionais (IF) nas quais decidimos se uma sequência de ações vai ser executada ou não em função da avaliação do valor de uma expressão booleana (as ações só serão executadas se a condição for verdadeira). E veja também a construção condicional onde usamos a condição para decidir entre duas sequências diferentes de ações (IF/ELSE), também em função do valor de uma expressão booleana.

- Estrutura Condicional

Nos exemplos onde introduzimos o IF e o IF/ELSE as ações em cada caso eram simples. Apenas um retorno do resultado da avaliação de alguma expressão. Veremos ao longo curso exemplos bem mais complexos. Para começar, nada impede que a sequência de ações a serem executadas quando a condição é verdadeira (ou falsa) inclua uma nova avaliação condicional. Por exemplo, quando vamos selecionando ações por eliminação. Por exemplo, para decidir se vou à praia, posso primeiro avaliar se está fazendo sol, e só então verificar se eu tenho dinheiro para a passagem de ônibus. Ou seja, o meu algoritmo nesse caso seria algo como, se está fazendo sol, então eu verifico minha carteira. Se eu tenho o dinheiro da passagem, então me arrumo e saio. Caso não esteja fazendo sol, eu vou ligar o computador, por exemplo, e nem vou olhar pra minha carteira. A implementação desse algoritmo pode ser realizada com um condicional dentro do outro (chamamos de *condicionais aninhados*). Em problemas complexos, onde os fatores a serem considerados para tomar uma decisão são

muitos, podemos ter muitos condicionais dentro de condicionais e o programa fica difícil de ser lido e entendido. Para esses casos, Python provê mais uma estrutura condicional, o IF/ELIF.

No vídeo a seguir, veja exemplos de funções utilizando essas diversas opções de estruturas condicionais:

- Exemplos utilizando Estruturas Condicionais

OBS: Os códigos das duas versões da função PosNegZero, mostradas no vídeo anterior, estão disponíveis como material de consulta na sua turma do Classroom.

- Função PosNegZero (implementação utilizando IF/ELSE aninhados)
- Função PosNegZero (implementação utilizando ELIF)

Atividade: Chegou a hora de testar os seus conhecimentos. Após terminar este exercício, responda às perguntas da atividade “Condicional”. Fique atento ao prazo de entrega dessa atividade!

Prática em Programação

Os conhecimentos deste roteiro serão o foco dos exercícios da prática dessa semana.
Atenção! Esta prática tem duas partes: uma para ser feita no IDLE, outra pra ser feita em uma ferramenta didática para ensino de computação, o Machine Teaching. Seu professor lhe dará os detalhes da entrega de suas resoluções.