

## PPT06 – Fundamentos de Python (UFCD 10793)

Sandra Liliana Meira de Oliveira



Coleções

# Tuplas

# Tuplas

Tuplas são usadas para armazenar vários itens em uma única variável.

Os itens de tupla são ordenados, imutáveis e permitem valores duplicados. Os itens de tupla são indexados, o primeiro item possui índice [0], o segundo item possui índice [1], etc...

- Ordenadas, significa que os itens têm uma ordem definida, e essa ordem não será alterada.
- Imutáveis, o que significa que não podemos alterar, adicionar ou remover itens após a criação da tupla.

# Tupla -Exemplo

```
t_fruta=("uva", "kiwi", "morango")
```

As tuplas são declaradas com parênteses e os elementos separados com virgulas.

# Tupla – Resumo de Operações

Resumo de Operações com Tuplas

# Sets

# Set

Sets não são ordenados, não podem ser alterados e não permitem valores duplicados.

- Não ordenado significa que os itens em um conjunto não têm uma ordem definida
- Os itens do conjunto podem aparecer em uma ordem diferente toda vez que você os usa e não podem ser referenciados por índice ou chave.



# Set -Exemplo

```
s_fruta={"uva", "kiwi", "morango"}
```

Os Sets são declaradas com chavetas e os elementos separados com virgulas.

## Operações com conjuntos - Set

# Set – Resumo de Operações

Método	Descrição	Exemplo
<code>set.add(x)</code>	Adiciona um elemento set	<pre>&gt;&gt;&gt; s = {1, 2, 3} &gt;&gt;&gt; s.add(4)      # {1, 2, 3, 4}</pre>
<code>set.clear()</code>	Remove todos os elementos do set	<pre>&gt;&gt;&gt; s = {1, 2, 3} &gt;&gt;&gt; s.clear()     # set()</pre>
<code>set.copy()</code>	Cria e retorna uma cópia do set	<pre>&gt;&gt;&gt; s = {1, 2, 'Alice'} &gt;&gt;&gt; s.copy()      # Returns: {1, 2, 'Alice'}</pre>
<code>set.difference(x)</code>	Retorna um novo conjunto com todos os elementos exceto os em x.	<pre>&gt;&gt;&gt; {1, 2, 3}.difference({1, 2}) {3}</pre>
<code>set.difference_update(iter)</code>	Remove todos os elemntos do set que estão no parâmetro.	<pre>&gt;&gt;&gt; s = {1, 2, 3} &gt;&gt;&gt; s.difference_update({1, 2}) # s == {3}</pre>
<code>set.discard(x)</code>	Remove um elemento do conjunto.	<pre>&gt;&gt;&gt; s = {'Alice', 'Bob', 'Cloe'} &gt;&gt;&gt; s.discard('Bob') # s == {'Alice', 'Cloe'}</pre>
<code>set.intersection()</code>	Retorna um novo set que resulta da interseção de ambos (elementos comuns)	<pre>&gt;&gt;&gt; {1, 2, 3, 4}.intersection({3, 4, 5}) {3, 4}</pre>
<code>set.intersection_update()</code>	Remove os elementos que não pertencem ao conjunto passado como parâmetro.	<pre>&gt;&gt;&gt; s = {1, 2, 3, 4} &gt;&gt;&gt; s.intersection_update({3, 4, 5}) # s == {3, 4}</pre>
<code>set.isdisjoint(x)</code>	Retorna verdadeiro se os dois sets não têm elementos em comum	<pre>&gt;&gt;&gt; {1, 2, 3, 4}.isdisjoint({'Alice', 'Bob'}) True</pre>
<code>set.issubset()</code>	Retorna verdadeiro se todos os elementos estão no conjunto passado.	<pre>&gt;&gt;&gt; t = {'Alice', 'Bob', 'Carl', 'Liz'} &gt;&gt;&gt; {'Alice', 'Bob'}.issubset(t) True</pre>
<code>set.issuperset()</code>	Retorna verdadeiro se todos os elementos do conjunto parametro estão no conjunto.	<pre>&gt;&gt;&gt; {'Alice', 'Bob', 'Carl'}.issuperset({'Alice'}) True</pre>
<code>set.pop()</code>	Remove e devolve um elemento do conjunto. Devolve erro se o set está vazio	<pre>&gt;&gt;&gt; s = {'Alice', 'Bob', 'Carl'} &gt;&gt;&gt; s.pop() 'Alice'</pre>
<code>set.remove()</code>	Remove e devolve um elemento específico do conjunto	<pre>&gt;&gt;&gt; s = {'Alice', 'Bob', 'Cloe'} &gt;&gt;&gt; s.remove('Bob') # s == {'Alice', 'Cloe'}</pre>
<code>set.symmetric_difference()</code>	Retorna a diferença entre conjuntos.	<pre>&gt;&gt;&gt; {1, 2, 3}.symmetric_difference({2, 3, 4}) {1, 4}</pre>
<code>set.symmetric_difference_update()</code>	Atualiza o conjunto com a diferença simétrica entre conjuntos.	<pre>&gt;&gt;&gt; s = {1, 2, 3} &gt;&gt;&gt; s.symmetric_difference_update({2, 3, 4}) &gt;&gt;&gt; s {1, 4}</pre>
<code>set.union()</code>	Cria e retorna um novo subset resultante da união dos dois.	<pre>&gt;&gt;&gt; {1, 2, 3, 4}.union({3, 4, 5}) {1, 2, 3, 4, 5}</pre>
<code>set.update()</code>	Atualiza o conjunto com os elementos que não estão no mesmo	<pre>&gt;&gt;&gt; s = {1, 2, 3, 4} &gt;&gt;&gt; s.update({3, 4, 5}) # s == {1, 2, 3, 4, 5}</pre>

## Operações em Conjuntos