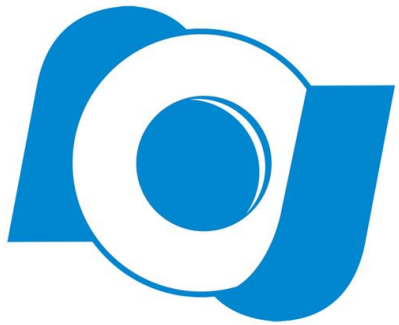
Tópicos de E-Science

2014 - 2

Martín Banda Huarca

Observatório Nacional



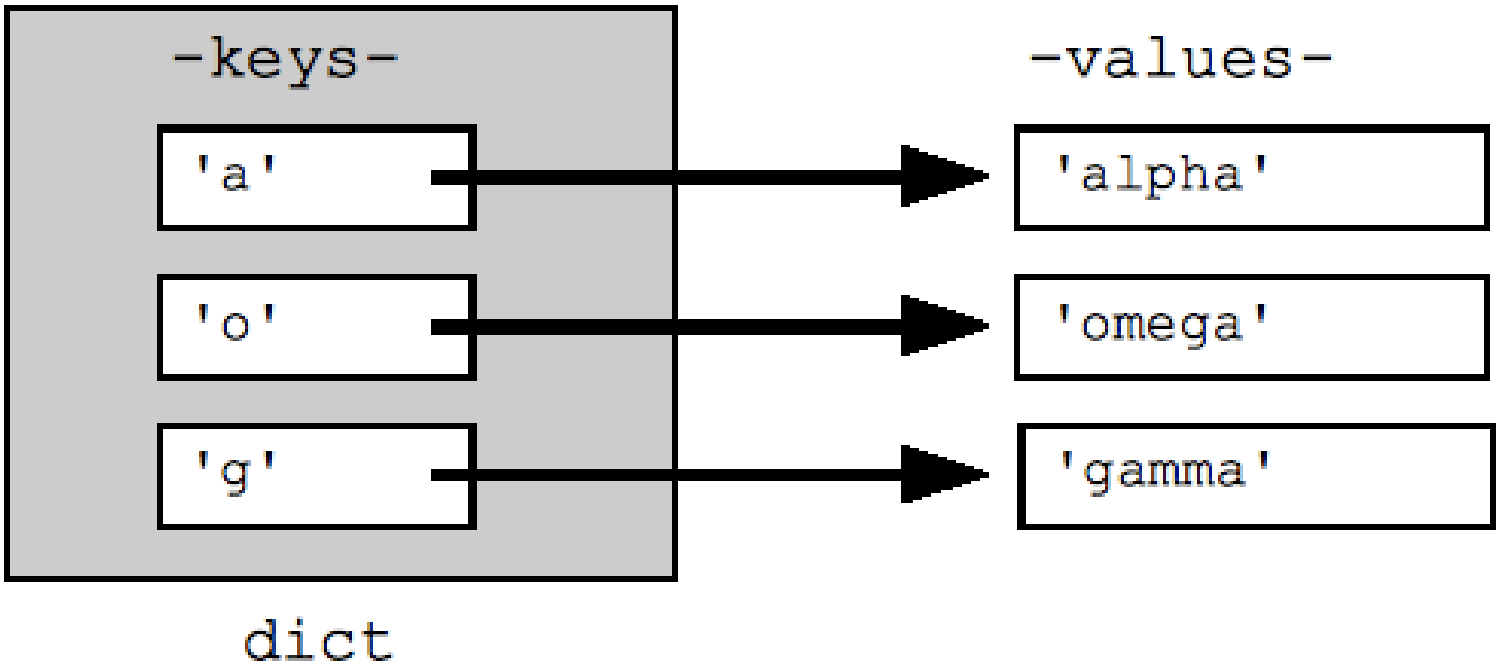
Rio de Janeiro

Conteúdo

Python dictionaries

Dicionário

Os dicionários em Python são estruturas de dados que contém pares de chave-valor.
São parecidos com as listas, mas o acesso a cada posição é feito de forma diferente.



Python dictionaries

Dicionário

Os dicionários do Python são também conhecidos como arrays associativos ou tabelas hash.

A sintaxe geral de um dicionário é da seguinte forma:

```
1 ramal = {'Cesar': '9160', 'Carlos': '9220', 'Martin
      ': '9223'}
```

```
1 dict1 = {'abc': 456}
2 dict2 = {'abc': 123, 98.6: 37}
```

Python dictionaries

Importante

- Cada chave é separada de seu valor por dois pontos (:).
- Os itens são separados por vírgulas.
- Todo o conjunto está entre chaves.
- Um dicionário vazio sem nenhum item é escrito com apenas duas chaves, assim: $\{\}$.
- As chaves são únicos dentro de um dicionário enquanto os valores não necessariamente.
- Os valores de um dicionário podem ser de qualquer tipo, mas as chaves devem ser de um tipo de dados imutáveis como strings, números ou tuplas.

Acessando valores no Dicionário

```
1 #!/usr/bin/python
2
3 dict = {'Name': 'Cesar', 'Age': 40, 'Class': 'E-
      Science'}
4
5 print "dict['Name']: ", dict['Name']
6 print "dict['Age']: ", dict['Age']
7 print "dict['Class']: ", dict['Class']
```

```
1 dict['Name']:  Cesar
2 dict['Age']:  40
3 dict['Class']: E-Science
```

Acessando valores no Dicionário

Warning

Se tentarmos acessar um item com uma chave, que não faz parte do dicionário, teremos um erro da seguinte forma:

```
1 #!/usr/bin/python
2
3 dict = {'Name': 'Andrea', 'Age': 24, 'Class': '
      Fisica'}
4
5 print "dict['Taysa']: ", dict['Taysa']
```

```
1 dict['Taysa']:
2 Traceback (most recent call last):
3   File "exemplo2.py", line 5, in <module>
4     print "dict['Taysa']: ", dict['Taysa']
5 KeyError: 'Taysa'
```

Atualizando o dicionário

```
1  #!/usr/bin/python
2
3  dict = {'Name': 'Martin', 'Age': 25, 'Class': '
       Astrometria'}
4
5  dict['Age'] = 20 # update existing entry
6  dict['Institute'] = "ON" # Add new entry
7
8
9  print "dict['Age']: ", dict['Age']
10 print "dict['School']: ", dict['School']
```

```
1  dict['Age']:  20
2  dict['School']:  ON
```

Removendo itens do dicionário

```
1  #!/usr/bin/python
2  dict = {'Name': 'Roberto', 'Age': 100, 'Class': '
       Mate'}
3
4  del dict['Name'] # remove entry with key 'Name'
5  print dict
6
7  dict.clear()     # remove all entries in dict
8  print dict
9
10 del dict         # delete entire dictionary
11 print dict
```

```
1 {'Age': 100, 'Class': 'Mate'}
2 {}
3 <type 'dict'>
```

Funções de um dicionário

Função	**Descrição**
cmp(dict1, dict2)	Compara elementos de ambos dict.
len(dict)	Dá a longitude total do dicionário. Este seria igual ao número de itens no dicionário.
str(dict)	Produces a printable string representation of a dictionary.
type(variable)	Retorna o tipo da variável dada. Se a variável dada é dicionário, em seguida, ele iria retornar um tipo dicionário.

Métodos de um dicionário

Método	**Descrição**
dict.clear()	Remove todos os elementos do dicionário dict
dict.copy()	Retorna uma cópia superficial do dicionário dict
dict.fromkeys()	Cria um novo dicionário com chaves de seq e valores set to value.
dict.get(key, default=None)	Para a chave key, retorna um valor ou padrão se a chave não está no dicionário
dict.has_key(key)	Retorna true se a chave está no dicionário dict, falso em caso contrário
dict.items()	Retorna uma lista de tupla pares (chave, valor)

Métodos de um dicionário

Método	**Descrição**
dict.keys()	retorna uma lista de chaves do dicionário dict
dict.setdefault(key, default=None)	Similar a get(), mas set dict[key]=default se a chave não está pronta em dict
dict.update(dict2)	Adiciona o par chave-valor de dict2 para o dicionário dict
dict.values()	Retorna lista de valores do dicionário dict

Conteúdo

Programação Orientado a Objetos

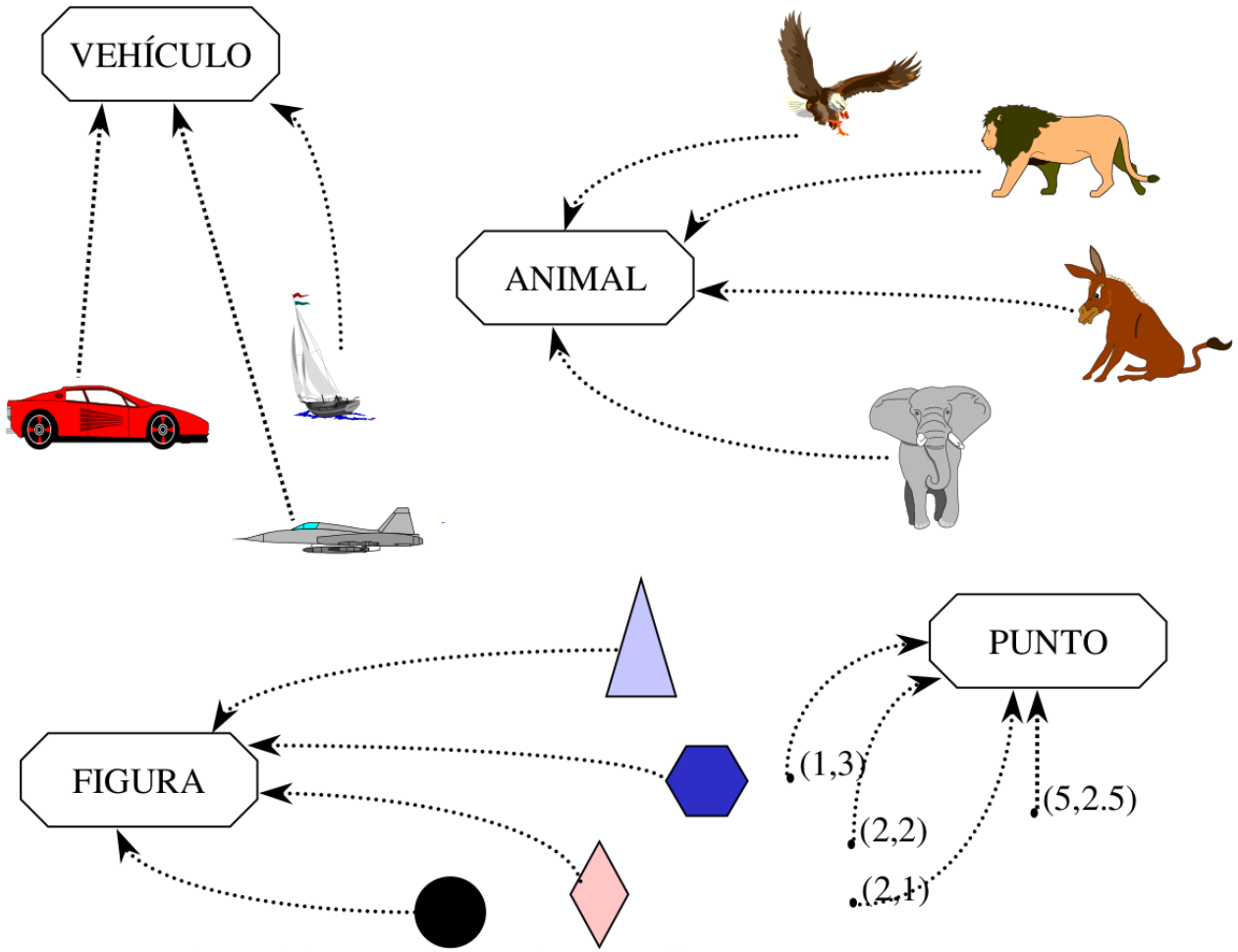
Definição

É um paradigma de programação que usa os objetos e as suas interações para desenvolver aplicações e programas informatizados.

Tipos de paradigmas de programação:

- Imperativo ou por procedimentos.
- Funcional.
- Lógico.
- Declarativo.
- Orientado a objetos.

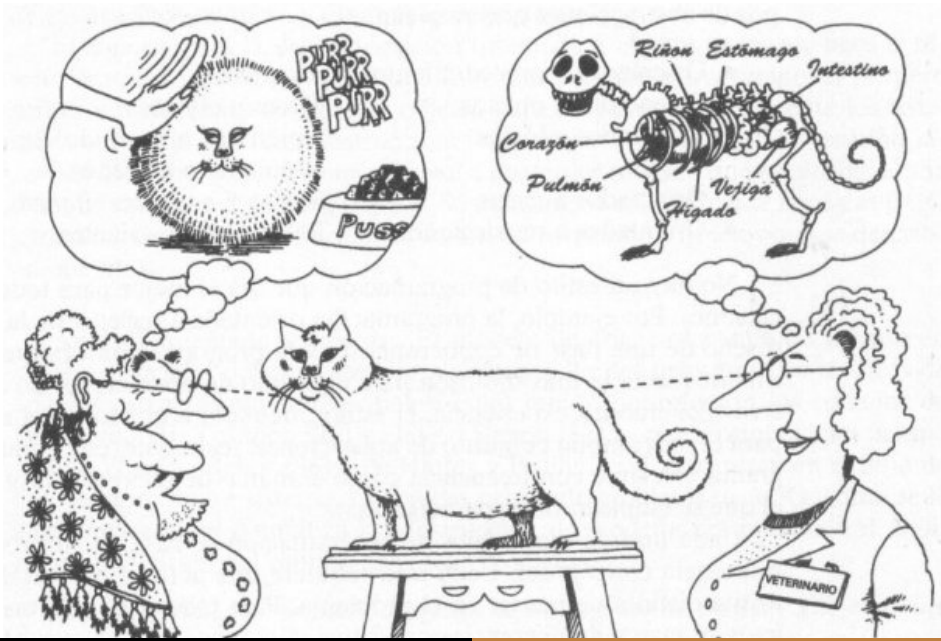
Exemplo de clases e objetos



Características da POO

Abstração

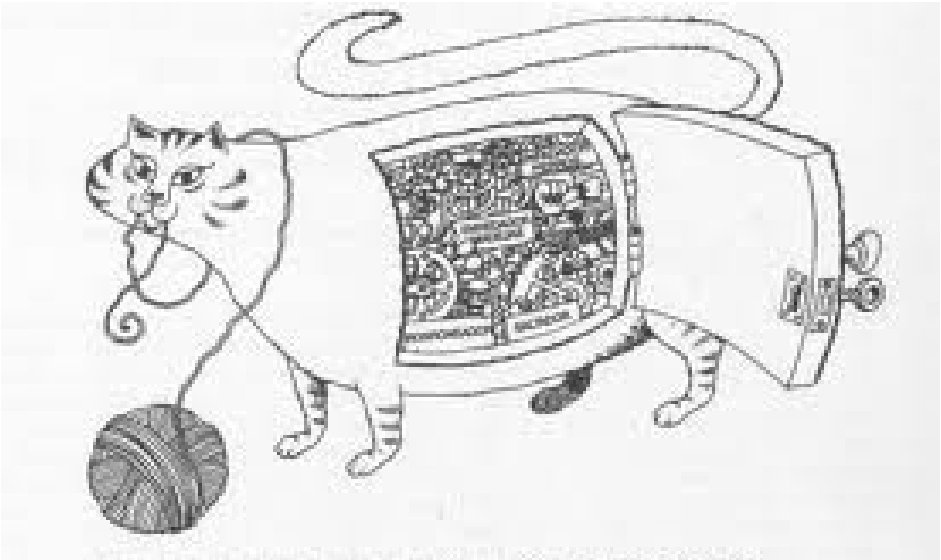
Denota as características essenciais de um objeto, onde os seus comportamentos são capturados. O processo de abstração permite selecionar as características relevantes do conjunto e identificar comportamentos comuns para definir novos tipos de entidades no mundo real.



Características da POO

Encapsulamento

Significa colocar todos os elementos que pertencem a mesma entidade no mesmo nível de abstração. Isto aumenta a coesão dos componentes do sistema.



No Zen of Python

Namespaces are one honking great idea - let's do more of those!

Namespaces são formas fundamentais e maravilhosas para organizar e encapsular variáveis e condições do ambiente em Python.

No Zen of Python

```cpp
#include <iostream>

int main()
{
   std::cout << "Hello World!";
}
```

```cpp
#include <iostream>

using namespace std;

int main()
{
   cout << "Hello World!";
}
```

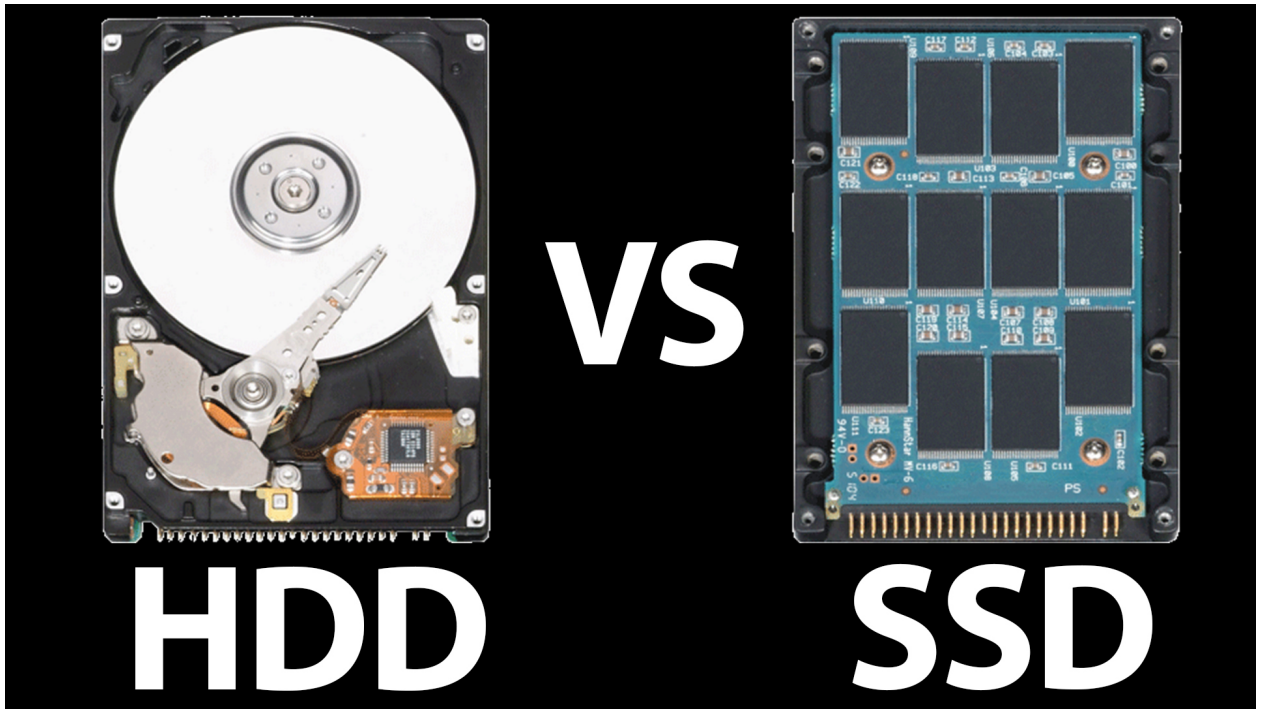
```
Hello World!
```

Conteúdo

SSD solid-state drive

SSD ou unidade de estado sólido

É um tipo de dispositivo, sem partes móveis, para armazenamento não volátil de dados digitais. São tipicamente construídos em torno de um circuito integrado semicondutor.



SSD solid-state drive

Funcionamento

Os SSDs mais comuns possuem dois componentes fundamentais: a memória flash e o controlador.

A memória flash

Salva todos os arquivos e, diferente dos discos magnéticos dos HDs, não necessita de partes móveis ou motores para funcionar.

SSD solid-state drive

SSD ou unidade de estado sólido

Todas as operações são feitas eletricamente, tornando as operações de leitura e escrita mais rápidas, além de deixar o drive mais silencioso e resistente a vibrações e quedas.

O crontolador

Gerencia a troca de dados entre o computador e a memória flash. Formado por um processador que executa diversas tarefas no drive, é um dos principais responsáveis pela performance de um SSD.

Vantagens

- Tempo de acesso reduzido. O tempo de acesso à memória é muito menor do que o tempo de acesso a meios magnéticos ou ópticos.
- Eliminação de partes móveis eletro-mecânicas, reduzindo vibrações, tornando-os completamente silenciosos.
- Por não possuírem partes móveis, são muito mais resistentes que os HDs comuns contra choques físicos.
- Menor peso em relação aos discos rígidos convencionais, mesmo os mais portáteis.
- Consumo reduzido de energia.
- Possibilidade de trabalhar em temperaturas maiores que os HDs comuns - cerca de $70^{\circ}C$.
- Largura de banda muito superior aos demais dispositivos, apresentando até $250MB/s$ na gravação e até $700MB/s$ nas operações de leitura.

Desvantagens

- Custo mais elevado.
- Capacidade de armazenamento inferior aos discos rígidos IDE e SATA.

OBRIGADO!!